APPENDIX H

SOFTWARE QUALITY ANALYSIS WORKSHEETS

H.1 Categories for Rating Compliance or Performance Levels

H.1.1 Poor Practice

This category represents the evaluation of modules having a compliance rating of 70%-79% of the criterion factor. A rating in this category, while not acceptable, may only require minimal efforts for re-engineering or repairs to effect a correction to the source code.

H.1.2 Acceptable

This category represents the evaluation of modules having a compliance rating of 80%-89% of the criterion factor. A rating in this category should not require any corrections, however, there can be recommendations for improvements to the source code.

H.1.3 Excellent

This category represents the evaluation of modules having a compliance rating of 90%-100% of the criterion factor. A rating in this category would not require any changes to the code, and there would be no recommendations for improvements due to the satisfaction of the evaluator. The non-applicable criterion are included in this category. For example: if a module did not utilize DO loops, the module would be considered 100% compliant with their use.

H.2 Performance Levels Points Assignments

Table H-1 shows the assignments of point values to performance levels. These values are to be assigned as a result of the evaluations of the sample set of modules. There are eighteen (18) criteria within the MOEs that will make up the points earned for each module. The maximum points that can be earned by a module is ninety (90).

Table H-1. Schedule of Points Assignments.

Performance Level	Points	Max Points
Poor Practice	1	
Acceptable	3	
Excellent	5	5 * 18 criteria = 90 points

H.3 Performance Level Evaluation Worksheets

The following filled-in worksheets are the Performance Level/Compliance evaluations for the 111 routines choosen as the sample for analysis.

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: acarea.f **Module Type**: Subroutine **Size:** 1471 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: adjenv.f **Module Type**: subroutine **Size:** 12,234 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations	✓		
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: aeval3 **Module Type**: subroutine **Size:** 16,959 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: aeval4.f **Module Type**: subroutine **Size:** 10,722 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: akshn1.f **Module Type**: subroutine **Size:** 4742 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: al344.f **Module Type**: subroutine **Size:** 4684 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: al347.f **Module Type**: subroutine **Size:** 6461 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: alt11.f **Module Type**: subroutine **Size:** 2994 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: alt12.f **Module Type**: subroutine **Size:** 3193 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: alt13.f **Module Type**: subroutine **Size:** 2558 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: alt31.f **Module Type**: subroutine **Size:** 3319 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: alt32.f **Module Type**: subroutine **Size:** 4946 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: alt33.f **Module Type**: subroutine **Size:** 5131 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: alt34.f **Module Type**: subroutine **Size:** 7927 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: alt35.f **Module Type**: subroutine **Size:** 8841 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: alt36.f **Module Type**: subroutine **Size:** 4115 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: alt37.f **Module Type**: subroutine **Size:** 3980 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: alt38.f **Module Type**: subroutine **Size:** 12,122 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: alt39.f **Module Type**: subroutine **Size:** 3397 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: alt3a.f **Module Type**: subroutine **Size:** 5123 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: alt3b.f **Module Type**: subroutine **Size:** 5463 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards	✓		
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices	✓		
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: alt3c.f **Module Type**: subroutine **Size:** 3352 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: alt3d.f **Module Type**: subroutine **Size:** 3352 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: alt3e.f **Module Type**: subroutine **Size:** 2596 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: alt3g.f **Module Type**: subroutine **Size:** 2187 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: alt3_17.f **Module Type**: subroutine **Size:** 5668 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: alt41.f **Module Type**: subroutine **Size:** 3573 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations	✓		
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: aoanow.f **Module Type**: subroutine **Size:** 9048 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: aproj3.f **Module Type**: subroutine **Size:** 26,863 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: aproj4.f **Module Type**: subroutine **Size:** 4638 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: aslct2.f **Module Type**: subroutine **Size:** 14,126 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: aslct3.f **Module Type**: subroutine **Size:** 7332 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: aslct4.f **Module Type**: subroutine **Size:** 5030 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: aslct7.f **Module Type**: subroutine **Size:** 5465 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: border.f **Module Type**: function **Size:** 2660 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations	✓		
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability	✓		

Module Name: bump.f **Module Type**: subroutine **Size:** 2732 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: canfir.f **Module Type**: subroutine **Size:** 3252 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: captrn.f **Module Type**: subroutine **Size:** 5095 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: chkwpn.f **Module Type**: subroutine **Size:** 3043 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: coninp.f **Module Type**: subroutine **Size:** 15,438 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability	✓		
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers	✓		
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability	✓		

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: cpzero.f **Module Type**: subroutine **Size:** 10,026 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations		✓	
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: convert85.f **Module Type**: subroutine **Size:** 146,748 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers	✓		
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices	✓		
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: cvrt_ir_exec.f **Module Type**: subroutine **Size:** 47,006 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices	✓		
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: disclientlib.c **Module Type**: subroutine **Size:** 70,948 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability			✓

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: dsmain.f **Module Type**: subroutine **Size:** 1446 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: d_msl_tkb.f **Module Type**: subroutine **Size:** 1817 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: eclin.f **Module Type**: subroutine **Size:** 4255 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices	✓		
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: effthr.f **Module Type**: subroutine **Size:** 4603 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: envlvl.f **Module Type**: subroutine **Size:** 8284 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: evbddr.f **Module Type**: subroutine **Size:** 5086 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: evdis.f **Module Type**: subroutine **Size:** 5234 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: evdrng.f **Module Type**: subroutine **Size:** 3493 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: evenvl.f **Module Type**: subroutine **Size:** 8898 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: evesc.f **Module Type**: subroutine **Size:** 1233 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability			✓

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: evevd.f **Module Type**: subroutine **Size:** 5439 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: evform.f **Module Type**: subroutine **Size:** 6794 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: evgrnd.f **Module Type**: subroutine **Size:** 9220 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: evilum.f **Module Type**: subroutine **Size:** 6181 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: evlosp.f **Module Type**: subroutine **Size:** 5594 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: evmaim.f **Module Type**: subroutine **Size:** 12,624 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity		✓	
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: evmain.f **Module Type**: subroutine **Size:** 1664 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: evmusp.f **Module Type**: subroutine **Size:** 6461 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements	✓		
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: evrte.f **Module Type**: subroutine **Size:** 4969 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity	✓		
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: evsep.f **Module Type**: subroutine **Size:** 7370 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability	✓		
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: evvec.f **Module Type**: subroutine **Size:** 4755 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements	✓		
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity		✓	
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: fcfire.f **Module Type**: subroutine **Size:** 5321 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: fcslt.f **Module Type**: subroutine **Size:** 4267 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: fctest.f **Module Type**: subroutine **Size:** 13,537 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: fflo.f **Module Type**: subroutine **Size:** 3854 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: flact0.f **Module Type**: subroutine **Size:** 3067 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: flact1.f **Module Type**: subroutine **Size:** 7202 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices	✓		
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: flaci.f **Module Type**: subroutine **Size:** 3012 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: flyac.f **Module Type**: subroutine **Size:** 12,637 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: gldmx.f **Module Type**: subroutine **Size:** 1941 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: gwreq.f **Module Type**: subroutine **Size:** 16,675 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: inputfile.f **Module Type**: subroutine **Size:** 14,902 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: minvrs.f **Module Type**: subroutine **Size:** 10,119 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers	✓		
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations		✓	
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability	✓		

Module Name: mislmd.f **Module Type**: subroutine **Size:** 12,407 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: mslenv.f5260 **Module Type**: subroutine **Size:** **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: msl_tkb_upd.f **Module Type**: subroutine **Size:** 38,744 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: msl_tk_fom.f **Module Type**: subroutine **Size:** 7056 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability	✓		

Module Name: ned_to_dis.c **Module Type**: subroutine **Size:** 28,035 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

BRAWLER V6.2 Software Analysis Worksheets.

Module Name: pnow.f **Module Type**: subroutine **Size:** 5456 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: prmnlm.f **Module Type**: subroutine **Size:** 3322 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: rates.f **Module Type**: subroutine **Size:** 9310 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: ratmx1.f **Module Type**: subroutine **Size:** 3480 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: ratmx2.f **Module Type**: subroutine **Size:** **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability			✓

Module Name: rdrelv.f **Module Type**: subroutine **Size:** 17,648 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: rline.f **Module Type**: subroutine **Size:** 12,849 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: rrrule.f **Module Type**: subroutine **Size:** 7370 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: rrsttv.f **Module Type**: subroutine **Size:** 14,812 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations	✓		
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: sanevt.f **Module Type**: subroutine **Size:** 12,840 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability	✓		

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: selrdr.f **Module Type**: subroutine **Size:** 13,002 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: selwpf.f **Module Type**: subroutine **Size:** 9019 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers	✓		
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: selwpi.f **Module Type**: subroutine **Size:** 8697 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability		✓	
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements		✓	
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: selwpn.f **Module Type**: subroutine **Size:** 7295 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: seteng.f **Module Type**: subroutine **Size:** 4417 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability			✓

Module Name: set_hldcod.f **Module Type**: subroutine **Size:** **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: shldfr.f **Module Type**: subroutine **Size:** 4084 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices	✓		
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: skrmod.f **Module Type**: subroutine **Size:** 5567 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability	✓		

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: skr_turn_on.f **Module Type**: subroutine **Size:** 13,851 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices	✓		
Criterion #4: Variable declarations		✓	
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability	✓		

Module Name: spbvri.f **Module Type**: subroutine **Size:** 5137 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: spbvrl.f **Module Type**: subroutine **Size:** 4425 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: thrlim.f **Module Type**: subroutine **Size:** 4802 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: tloss.f **Module Type**: subroutine **Size:** 2228 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations	✓		
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability			✓

Module Name: updtev.f **Module Type**: subroutine **Size:** 13,850 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: visobs.f **Module Type**: subroutine **Size:** 9030 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity		✓	
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: vscan.f **Module Type**: subroutine **Size:** 13,200 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability			✓
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: wptgpr.f **Module Type**: subroutine **Size:** 13,132 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers			✓
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

Module Name: xlift.f **Module Type**: subroutine **Size:** 3509 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards		✓	
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	

TAC BRAWLER V6.2 Software Analysis Worksheets.

Module Name: xyz_to_sphere.f **Module Type**: subroutine **Size:** 7767 **bytes**

Criterion	Poor Practice	Acceptable	Excellent
MOE #1 - Use of Standards:			
Criterion #1: Readability		✓	
Criterion #2: Modifiability			✓
Criterion #3: ANSI standards			✓
MOE #2 - Programming Conventions:			
Criterion #1: Use of comments and headers		✓	
Criterion #2: Use of formatted statements			✓
Criterion #3: Logical I/O devices			✓
Criterion #4: Variable declarations			✓
Criterion #5: Variable initialization			✓
Criterion #6: Variable naming conventions			✓
Criterion #7: Algorithm clarity			✓
MOE #3 - Computational Efficiency:			
Criterion #1: Mixed mode calculations			✓
Criterion #2: Use of library functions			✓
Criterion #3: Nested computations			✓
MOE #4 - Maintainability:			
Criterion #1: Portability			✓
Criterion #2: Memory management			✓
Criterion #3: Use of COMMON blocks			✓
Criterion #4: Modularity			✓
Criterion #5: Subroutine traceability		✓	